WE CLAIM:

1. A CDR (clock data recovery) deserializer comprising:

a clock divider that receives a recovered clock signal having a first frequency and that generates a divided clock signal having a second frequency, wherein said second frequency of said divided clock signal is lowered from said first frequency of said recovered clock signal by a predetermined ratio number such that each cycle of said divided clock signal is generated for each count of cycles of said recovered clock signal up to said predetermined ratio number;

a serial-to-parallel shift register that shifts in recovered serial data bits with each cycle of said recovered clock signal and outputs, as RPDO (recovered parallel data output), said predetermined ratio number of said shifted recovered serial data bits at a predetermined transition of every cycle of said divided clock signal;

a SYNC (synchronization) detect logic for asserting a VRS (diVider ReSet) signal coupled to said clock divider for controlling said clock divider to generate said predetermined transition for a cycle of said divided clock signal when said VRS signal is asserted, wherein said SYNC detect logic includes:

a first reloadable register portion for storing a first synchronization bit pattern comprised of a first predetermined number of bits for a first communications protocol, wherein said first reloadable register portion is capable of being coupled to a first port for inputting said first synchronization bit pattern that is programmed into said first reloadable register portion through said first port;

a second reloadable register portion for storing a second synchronization bit pattern comprised of a second predetermined number of bits for a second communications protocol, wherein said second reloadable register portion is capable of being coupled to a second port for inputting said second synchronization bit pattern that is programmed into said second

reloadable register portion through said second port;

a first bit pattern comparator that inputs an intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal, wherein said serial-to-parallel shift register shifts in a bit of said recovered serial data bits every cycle of said recovered clock signal to generate said intermediate parallel data output (IPDO), and wherein said first bit pattern comparator compares for every cycle of said recovered clock signal said first predetermined number of bits of said intermediate parallel data output (IPDO) to said first synchronization bit pattern stored within said first reloadable register portion to assert a first comparator output signal when said first predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said first synchronization bit pattern;

a second bit pattern comparator that inputs said intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal and that compares for every cycle of said recovered clock signal said second predetermined number of bits of said intermediate parallel data output (IPDO) to said second synchronization bit pattern stored within said second reloadable register portion to assert a second comparator output signal when said second predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said second synchronization bit pattern; and

a multiplexer that inputs said first and second comparator output signals and that outputs said VRS signal coupled to said clock divider, wherein said multiplexer selects said first comparator output signal as said VRS signal when said recovered serial data bits are for said first communications protocol, and wherein said multiplexer selects said second comparator output signal as said VRS signal when said recovered serial data

bits are for said second communications protocol.

2. The CDR (clock data recovery) deserializer of claim 1, wherein said SYNC (synchronization) detect logic further includes:

a third reloadable register portion for storing a third synchronization bit pattern comprised of a third predetermined number of bits for said second communications protocol, wherein said third reloadable register portion is capable of being coupled to a third port for inputting said third synchronization bit pattern that is programmed into said third reloadable register portion through said third port;

a third bit pattern comparator that inputs said intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal and that compares for every cycle of said recovered clock signal said third predetermined number of bits of said intermediate parallel data output (IPDO) to said third synchronization bit pattern stored within said third reloadable register portion to assert a third comparator output signal when said third predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said third synchronization bit pattern; and

a combinational logic that inputs said second comparator output signal and said third comparator output signal to generate a combinational logic output signal that is input to said multiplexer, wherein said multiplexer selects said combinational logic output signal as said VRS signal when said recovered serial data bits are for said second communications protocol.

3. The CDR (clock data recovery) deserializer of claim 2, wherein said combinational logic is comprised of an OR-gate.

4. The CDR (clock data recovery) deserializer of claim 2, wherein a computer

system is used for programming said first, second, and third synchronization bit patterns into said first, second, and third reloadable register portions, respectively, by software from said computer system.

5. The CDR (clock data recovery) deserializer of claim 2, wherein said first, second, and third reloadable register portions are each a part of a single reloadable register.

6. The CDR (clock data recovery) deserializer of claim 2, wherein said first, second, and third reloadable register portions are each a part of a separate respective reloadable register.

7. The CDR (clock data recovery) deserializer of claim 1, wherein a computer system is used for programming said first and second synchronization bit patterns into said first and second reloadable register portions, respectively, by software from said computer system.

8. The CDR (clock data recovery) deserializer of claim 1, wherein said CDR (clock data recovery) deserializer is within a SERDES (serializer/deserializer) transceiver.

9. A CDR (clock data recovery) deserializer comprising:

a clock divider that receives a recovered clock signal having a first frequency and that generates a divided clock signal having a second frequency, wherein said second frequency of said divided clock signal is lowered from said first frequency of said recovered clock signal by a predetermined ratio number such that each cycle of said divided clock signal is generated for each count of cycles of said recovered clock signal up to said predetermined ratio number;

a serial-to-parallel shift register that shifts in recovered serial data bits with

each cycle of said recovered clock signal and outputs, as RPDO (recovered parallel data output), said predetermined ratio number of said shifted recovered serial data bits at a predetermined transition of every cycle of said divided clock signal;

a SYNC (synchronization) detect logic for asserting a VRS (diVider ReSet) signal coupled to said clock divider for controlling said clock divider to generate said predetermined transition for a cycle of said divided clock signal when said VRS signal is asserted, wherein said SYNC detect logic includes:

means for storing a first synchronization bit pattern comprised of a first predetermined number of bits for a first communications protocol and programmed through a first port;

means for storing a second synchronization bit pattern comprised of a second predetermined number of bits for a second communications protocol and programmed through a second port;

means for inputting an intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal, wherein said serial-to-parallel shift register shifts in a bit of said recovered serial data bits every cycle of said recovered clock signal to generate said intermediate parallel data output (IPDO), and means for comparing for every cycle of said recovered clock signal said first predetermined number of bits of said intermediate parallel data output (IPDO) to said first synchronization bit pattern to assert a first comparator output signal when said first predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said first synchronization bit pattern;

means for inputting said intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal and for comparing for every cycle of said recovered clock signal said

second predetermined number of bits of said intermediate parallel data output (IPDO) to said second synchronization bit pattern to assert a second comparator output signal when said second predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said second synchronization bit pattern; and

means for selecting one of said first and second comparator output signals as said VRS signal coupled to said clock divider, wherein said first comparator output signal is selected as said VRS signal when said recovered serial data bits are for said first communications protocol, and wherein said second comparator output signal is selected as said VRS signal when said recovered serial data bits are for said second communications protocol.

10. The CDR (clock data recovery) deserializer of claim 9, wherein said SYNC (synchronization) detect logic further includes:

means for storing a third synchronization bit pattern comprised of a third predetermined number of bits for said second communications protocol and programmed through a third port;

means for inputting said intermediate parallel data output (IPDO) from said serial-to-parallel shift register with each cycle of said recovered clock signal and for comparing for every cycle of said recovered clock signal said third predetermined number of bits of said intermediate parallel data output (IPDO) to said third synchronization bit pattern to assert a third comparator output signal when said third predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said third synchronization bit pattern;

a combinational logic that inputs said second comparator output signal and said third comparator output signal to generate a combinational logic output signal; and

means for selecting said combinational logic output signal as said VRS signal when said recovered serial data bits are for said second communications protocol.

11. The CDR (clock data recovery) deserializer of claim 10, wherein said combinational logic is comprised of an OR-gate.

12. The CDR (clock data recovery) deserializer of claim 10, further comprising:

means for programming said first, second, and third synchronization bit patterns through said first, second, and third port, respectively.

13. The CDR (clock data recovery) deserializer of claim 10, wherein said first, second, and third reloadable register portions are each a part of a single reloadable register.

14. The CDR (clock data recovery) deserializer of claim 10, wherein said first, second, and third reloadable register portions are each a part of a separate respective reloadable register.

15. The CDR (clock data recovery) deserializer of claim 9, further comprising:

means for programming said first and second synchronization bit patterns through said first and second ports, respectively.

16. The CDR (clock data recovery) deserializer of claim 9, wherein said CDR (clock data recovery) deserializer is within a SERDES (serializer/deserializer) transceiver.

17. A method for deserializing recovered serial data bits with a recovered clock signal, the method comprising:

generating a divided clock signal having a second frequency from said

recovered clock signal having a first frequency, wherein said second frequency of said divided clock signal is lowered from said first frequency of said recovered clock signal by a predetermined ratio number such that each cycle of said divided clock signal is generated for each count of cycles of said recovered clock signal up to said predetermined ratio number;

shifting in recovered serial data bits with each cycle of said recovered clock signal and outputting, as RPDO (recovered parallel data output), said predetermined ratio number of said shifted recovered serial data bits at a predetermined transition of every cycle of said divided clock signal; and

asserting a VRS (diVider ReSet) signal to generate said predetermined transition for a cycle of said divided clock signal when said VRS signal is asserted, said step of asserting said VRS signal further including:

programming a first synchronization bit pattern comprised of a first predetermined number of bits for a first communications protocol into a first reloadable register portion coupled to a first port for inputting said first synchronization bit pattern that is programmed into said first reloadable register portion through said first port;

programming a second synchronization bit pattern comprised of a second predetermined number of bits for a second communications protocol into a second reloadable register portion coupled to a second port for inputting said second synchronization bit pattern that is programmed into said second reloadable register portion through said second port;

inputting an intermediate parallel data output (IPDO) with each cycle of said recovered clock signal, wherein a bit of said recovered serial data bits is shifted in every cycle of said recovered clock signal to generate said intermediate parallel data output (IPDO), and comparing for every cycle of said recovered clock signal said first predetermined number of bits of said

intermediate parallel data output (IPDO) to said first synchronization bit pattern stored within said first reloadable register portion to assert a first comparator output signal when said first predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said first synchronization bit pattern;

inputting said intermediate parallel data output (IPDO) with each cycle of said recovered clock signal and comparing for every cycle of said recovered clock signal said second predetermined number of bits of said intermediate parallel data output (IPDO) to said second synchronization bit pattern stored within said second reloadable register portion to assert a second comparator output signal when said second predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said second synchronization bit pattern; and

selecting said first comparator output signal as said VRS signal when said recovered serial data bits are for said first communications protocol, and selecting said second comparator output signal as said VRS signal when said recovered serial data bits are for said second communications protocol.

18. The method of claim 17, wherein said step of asserting said VRS signal further includes:

programming a third synchronization bit pattern comprised of a third predetermined number of bits for said second communications protocol into a third reloadable register portion coupled to a third port for inputting said third synchronization bit pattern that is programmed into said third reloadable register portion through said third port;

inputting said intermediate parallel data output (IPDO) with each cycle of said recovered clock signal and comparing for every cycle of said recovered clock signal

said third predetermined number of bits of said intermediate parallel data output (IPDO) to said third synchronization bit pattern stored within said third reloadable register portion to assert a third comparator output signal when said third predetermined number of bits of said intermediate parallel data output (IPDO) is substantially same as said third synchronization bit pattern;

inputting said second comparator output signal and said third comparator output signal into a combinational logic to generate a combinational logic output signal; and

selecting said combinational logic output signal as said VRS signal when said recovered serial data bits are for said second communications protocol.

19. The method of claim 18, wherein said combinational logic is comprised of an OR-gate.

20. The method of claim 18, further comprising:

programming said first, second, and third synchronization bit patterns into said first, second, and third reloadable register portions, respectively, by software from a computer system.

21. The method of claim 18, wherein said first, second, and third reloadable register portions are each a part of a single reloadable register.

22. The method of claim 18, wherein said first, second, and third reloadable register portions are each a part of a separate respective reloadable register.

23. The method of claim 17, further comprising:

programming said first and second synchronization bit patterns into said first

and second reloadable register portions, respectively, by software from a computer system.

24. A CDR (clock data recovery) deserializer comprising:

a clock divider operable to receive a recovered clock signal (SCLK) and to generate a divided clock signal (RPCLK) for every given number of recovered clock signal cycles;

a serial-to-parallel shift register responsive to the recovered clock signal to shift in recovered serial data bits and responsive to the divided clock signal to shift out recovered parallel data bits; and

sync detect logic coupled to the clock divider and operable to assert a parallel clock enabling signal that enables the clock divider to generate the divided clock signal, the sync detect logic including a reloadable register operable to store a programmable synchronization bit pattern associated with a communications protocol and a bit pattern comparator operable to compare the stored programmable synchronization bit pattern with a synchronization bit pattern within the recovered serial data bits and to assert or not assert the parallel clock enabling signal as a result of the comparison.

25. The CDR (clock data recovery) deserializer of claim 24, further comprising a DPLL (digital phase locked loop) for generating the recovered clock signal, the DPLL including:

a phase detector operable to compare a serial data input (SDIN) with a recovered clock signal (SCLK) and to generate in response up and down signals;

a phase selector operable to select a clock signal as the recovered clock signal (SCLK) from a plurality of given clock signals in response to FWD (forward) and BWD (backward) signals; and

a digital filter coupled between the phase detector and the phase selector, the digital filter operable to generate the FWD and BWD signals for the phase selector in response to the up and down signals received from the phase detector;

wherein the digital filter includes at least one reloadable register operable to store a programmable value for comparison with a value derived from the up and down signals and a controller responsive to the comparison and operable to generate the FWD and BWD signals.

26. The CDR (clock data recovery) deserializer of claim 24, further comprising a DPLL (digital phase locked loop) for generating the recovered clock signal, the DPLL including:

a phase detector operable to compare a serial data input (SDIN) with a recovered clock signal (SCLK) and to generate in response up and down signals;

a digital filter coupled to the phase detector and operable to generate FWD (forward) and BWD (backward) signals in response to the up and down signals; and

a phase selector coupled to the digital filter and including a phase interpolator coupled to a multiplexer responsive to the FWD and BWD signals, the multiplexer operable to receive a plurality of given clock signals having different phases as inputs and to select each of at least two clock signals as one of the given clock signals as outputs, the phase interpolator operable to generate a recovered clock signal (SCLK) having a phase that is phase interpolated between the phases of the at least two selected clock signals.

27. The CDR (clock data recovery) deserializer of claim 24, further comprising a DPLL (digital phase locked loop) for generating the recovered clock signal, the DPLL including:

a phase detector operable to compare a serial data input (SDIN) with a

recovered clock signal (SCLK) and to generate in response up and down signals;

a digital filter coupled to the phase detector and operable to generate FWD (forward) and BWD (backward) signals in response to the up and down signals;

wherein the digital filter includes at least one reloadable register operable to store a programmable value for comparison with a value derived from the up and down signals and a controller responsive to the comparison and operable to generate the FWD and BWD signals; and

a phase selector coupled to the digital filter and including a phase interpolator coupled to a multiplexer responsive to the FWD and BWD signals, the multiplexer operable to receive a plurality of given clock signals having different phases as inputs and to select each of at least two clock signals as one of the given clock signals as outputs, the phase interpolator operable to generate a recovered clock signal (SCLK) having a phase that is phase interpolated between the phases of the at least two selected clock signals.

28. The CDR deserializer of claim 24, wherein the CDR deserializer is part of a SERDES (serializer/deserializer) transceiver.

29. The CDR deserializer of claim 24 including at least two reloadable registers, each operable to store a programmable synchronization bit pattern associated with a communications protocol.

30. The CDR deserializer of claim 29 including means for selecting between the stored programmable synchronization bit patterns in asserting or not asserting the parallel clock enabling signal.

31. Sync detect logic comprising:

at least two reloadable registers, each operable to store a programmable synchronization bit pattern associated with a communications protocol;

a bit comparator operable to compare a selected programmable synchronization bit pattern with a synchronization bit pattern within recovered serial data bits and to assert or not assert a parallel clock enabling signal as a result of the comparison; and

means for selecting between the stored programmable synchronization bit patterns in asserting or not asserting the parallel clock enabling signal.

32. The sync detect logic of claim 31, wherein said selecting means is comprised of a multiplexer disposed after a plurality of bit comparators with said multiplexer selecting one of the outputs of the bit comparators as said parallel clock enabling signal depending on the communications protocol indicated by a MODE signal, and wherein each of said bit comparators compares a respective synchronization bit pattern to said synchronization bit pattern within said recovered serial data bits.

33. The sync detect logic of claim 31, wherein said selecting means is comprised of a multiplexer disposed before a bit comparator and after said at least two reloadable registers with said multiplexer selecting one of the programmable synchronization bit patterns from the at least two reloadable registers depending on the communications protocol indicated by a MODE signal, and wherein said bit comparator compares said selected programmable synchronization bit pattern to said synchronization bit pattern within recovered serial data bits to generate said parallel clock enabling signal.

34. A method of deserializing recovered serial data bits with a recovered clock signal comprising:

generating a divided clock signal (RPCLK) in response to a given number of

recovered clock signal cycles and the assertion of a parallel clock enabling signal;

shifting in recovered serial data bits in response to the recovered clock signal and shifting out recovered parallel data bits in response to the divided clock signal;

comparing a programmable synchronization bit pattern with a synchronization bit pattern within the recovered serial data bits; and

asserting or not asserting the parallel clock enabling signal as a result of the comparison.

35. The method of claim 34 including:

providing at least two programmable synchronization bit patterns; and

selecting between the programmable synchronization bit patterns in asserting or not asserting the parallel clock enabling signal.

36. The method of claim 35, wherein selecting between the programmable synchronization bit patterns in asserting or not asserting the parallel clock enabling signal includes comparing each of the programmable synchronization bit patterns with the synchronization bit pattern within the recovered serial data bits and then selecting the results of the comparison.

37. The method of claim 35, wherein selecting between the programmable synchronization bit patterns in asserting or not asserting the parallel clock enabling signal includes selecting between the programmable synchronization bit patterns and then comparing the selected programmable synchronization bit pattern with the synchronization bit pattern within the recovered serial data bits.